

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-30455

(43) 公開日 平成8年(1996)2月2日

(51) Int.Cl.⁶

G 0 6 F 9/44
9/46

識別記号

5 3 0 M 7737-5B
3 4 0 A 7737-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数 4 O L (全 14 頁)

(21) 出願番号 特願平6-164996

(22) 出願日 平成6年(1994)7月18日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 横山 孝典

茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内

(72) 発明者 島田 優

茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内

(72) 発明者 齊藤 雅彦

茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内

(74) 代理人 弁理士 高橋 明夫 (外1名)

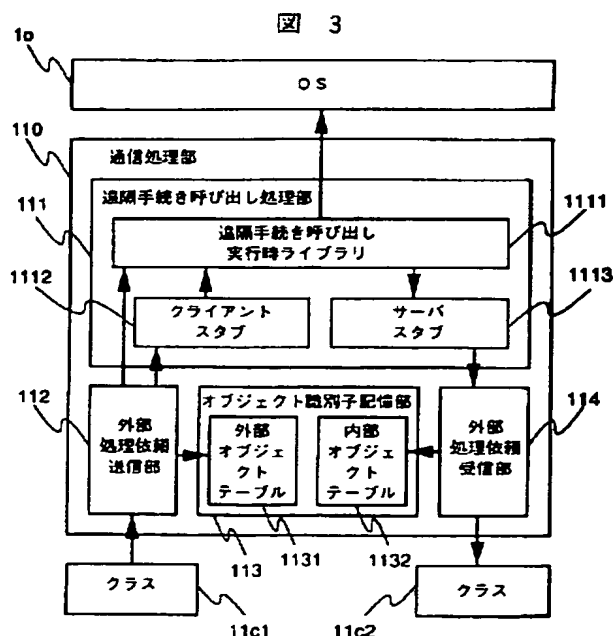
最終頁に続く

(54) 【発明の名称】 オブジェクト指向情報処理システム

(57) 【要約】

【目的】複数のプロセスのオブジェクト間で処理を依頼し合う場合において、プロセス内のオブジェクトについては、オブジェクトアドレス指定により、効率を良く高速に処理する。また、アプリケーションプログラマが、システムを容易に構築できるようにして、プログラムの再利用や移植も容易にする。

【構成】オブジェクトアドレスとオブジェクト識別子との両者の対応を記憶するオブジェクト識別子記憶手段113と、外部処理依頼送信手段112と、外部処理依頼受信手段114とを設け、外部プロセスにメッセージ送信をするときに、外部処理依頼送信手段は、オブジェクトアドレスを対応するオブジェクト識別子に変換して送信する。受信側では、外部処理依頼受信手段が、受信メッセージに含まれオブジェクト識別子を対応する処理依頼先オブジェクトのオブジェクトアドレスに変換して、このメッセージを受信したプロセス内で以後の処理を進める。



【特許請求の範囲】

【請求項 1】 複数のオブジェクトを含むプロセスが複数存在し、前記オブジェクト間で処理を依頼し合って処理を進めるオブジェクト指向情報処理システムにおいて、

各々のオブジェクトに、
プロセス内で、オブジェクトを一意に指定するためのオブジェクトアドレスと、このオブジェクト指向情報処理システム全体で、オブジェクトを一意に指定するためのオブジェクト識別子とを設け、

各々のプロセス内に、
前記オブジェクトと前記オブジェクト識別子の対応を記憶するオブジェクト識別子記憶手段と、
外部処理依頼送信手段と、

外部処理依頼受信手段とを有し、
処理依頼元オブジェクトがその処理依頼元オブジェクトが存在するプロセスとは異なるプロセスに存在する処理依頼先オブジェクトに処理を依頼する場合に、

前記外部処理依頼送信手段が、前記オブジェクト識別子記憶手段を参照して、オブジェクトアドレスを、対応するオブジェクト識別子に変換し、そのオブジェクト識別子を指定して、処理依頼メッセージを依頼先プロセスに送信して、

前記外部処理依頼受信手段が、外部のプロセスの処理依頼元オブジェクトからの処理依頼メッセージを受信した場合に、前記オブジェクト識別子記憶手段を参照して、前記受信した処理依頼メッセージで指定されたオブジェクト識別子を、対応する処理依頼先オブジェクトのオブジェクトアドレスに変換して、

そのオブジェクトアドレスを参照することにより、このメッセージを受信したプロセス内で処理を進めることを特徴とするオブジェクト指向情報処理システム。

【請求項 2】 前記外部処理依頼送信手段は、処理依頼メッセージに付加された引数に含まれるオブジェクトアドレスについて、前記オブジェクト識別子記憶手段を参照して、前記オブジェクトアドレスを、対応するオブジェクト識別子に変換して処理依頼メッセージを送信して、

前記外部処理依頼受信手段は、前記受信した処理依頼メッセージ中の引数に含まれた前記オブジェクト識別子を、対応する処理依頼先オブジェクトのオブジェクトアドレスに変換して、

そのオブジェクトアドレスを参照することにより、このメッセージを受信したプロセス内で処理を進めることを特徴とする請求項 1 記載のオブジェクト指向情報処理システム。

【請求項 3】 前記外部処理依頼送信手段は、処理依頼メッセージに付加された引数に含まれるオブジェクトアドレスについて、前記オブジェクト識別子記憶手段を参照して、前記オブジェクトアドレスを、対応するオブジ

ェクト識別子に変換して処理依頼メッセージを送信して、

前記外部処理依頼受信手段は、処理依頼メッセージ中の引数に含まれたオブジェクト識別子のうち、この処理メッセージを受信したプロセス上に存在するオブジェクトに対応するオブジェクト識別子についてのみ、前記オブジェクト識別子記憶手段を参照して、そのオブジェクト識別子を、対応する処理依頼先オブジェクトのオブジェクトアドレスに変換して、

10 そのオブジェクトアドレスを参照することにより、このメッセージを受信したプロセス内で処理を進めることを特徴とする請求項 1 記載のオブジェクト指向情報処理システム。

【請求項 4】 前記オブジェクト識別子は、前記オブジェクトアドレスと同一のデータ型とすることを特徴とする請求項 1 ないし請求項 3 記載のいずれかのオブジェクト指向情報処理システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、オブジェクト指向情報処理システムに係り、複数のオブジェクトが処理を依頼し合い、しかも、その複数のオブジェクトが存在するプロセスが複数存在するオブジェクト指向情報処理システムであって、特に、処理の記述を容易にして、プロセスを意識せずに処理の記述をおこないうるオブジェクト指向情報処理システムに関する。

【0002】

【従来の技術】従来、一つのプロセス中に複数のオブジェクトが存在し処理をおこなうシステムでは、オブジェクトをプロセス内で一意に指定できる識別子を用いて指定する方法が採られていた。この方法としては、オブジェクトのデータが存在するメモリ上のアドレスを用いて識別する方法が一般的である。

【0003】一般に、オブジェクトへ処理を依頼する場合には、処理を手続き（サブルーチン、関数）呼び出しにより実行する。そして、上記の方法では、処理を依頼するオブジェクトの指定を、そのオブジェクトの記憶空間上のアドレスを用いておこない、手続き呼び出しの第 1 引数でそのアドレスを指定するものである。

【0004】ところが、アドレスはプロセス内では一意にオブジェクトを指定できるが、他のプロセスについては、メモリ空間が通常は、異なるため一意に指定できず、上記方法では、複数のプロセス上のオブジェクト間で処理を依頼し合うことができない。

【0005】そこで、例えば、エム・アイ・ティー・プレス社（The MIT Press）刊、「A B C L オブジェクト指向コンカレント システム」（1990 年）（ABCL AnObject-Oriented Concurrent System（1990））、133 頁から 155 頁に記載されているように、システム全体が複数の計算機を有する場合であっても、システム全体

20

30

40

50

で一意にオブジェクトを指定できる識別子を用い、処理依頼元オブジェクトは、処理依頼先オブジェクトが処理依頼元オブジェクトと同一の計算機に存在するか、他の計算機に存在するかに関わらず、上記識別子により依頼先オブジェクトを指定し、処理依頼時に依頼先オブジェクトがどの計算機に存在するかを調べ、その計算機上のオブジェクトに処理を依頼する方法がある。なお、この文献では、計算機ごとにメモリ空間が異なるが、これは通常、プロセスが異なると、メモリ空間が異なることに

【0006】また、特開平5-12025号公報に記載されているように、他のプロセスに存在する処理依頼先オブジェクトに対応する仮想的なオブジェクトを、処理依頼元オブジェクトが存在するプロセス内に設け、処理依頼元オブジェクトが、同一プロセス内の仮想的な処理依頼先オブジェクトに処理を依頼すると、その仮想的な処理依頼先オブジェクトは、他のプロセスに存在する処理依頼先オブジェクトに対して通信をおこない、その処理依頼先オブジェクトが処理を実行することにより、複数のプロセス上のオブジェクト間で処理を依頼し合うこ

【0007】

【発明が解決しようとする課題】上記従来技術は、複数のオブジェクトが複数のプロセスに分散しているときに、互いの通信を可能にするものである。しかしながら、上記第一の従来技術では、他のオブジェクトへの処理依頼発生のたびに、処理依頼先のオブジェクトが同一プロセス（計算機）内に存在するかどうかを調べる必要があるため、処理に時間がかかるうえ、システム内のオブジェクト全てが同一処理系で実行されなければならないという制限があるという問題があった。

【0008】また、上記第二の従来技術では、アプリケーションプログラマは、他のプロセスに存在する処理依頼先オブジェクトについて、仮想的なオブジェクトを設け、上記他のプロセスのオブジェクトとの通信処理を記述しなければならないという煩わしさがあ

【0009】本発明は、上記問題点を解決するためになされたもので、その目的は、複数のプロセスのオブジェクト間で処理を依頼し合う場合において、プロセス内のオブジェクトについてはオブジェクトアドレス指定により高速に処理することでき、効率のよいオブジェクト指向情報処理システムを提供することにある。

【0010】またその目的は、上記場合において、アプリケーションプログラマが、他のプロセス上の処理依頼先オブジェクトとの通信処理を行うためのプログラムを記述する必要がなく、複数のプロセス上で通信するオブジェクト指向情報処理システムを、容易に構築することが可能であり、しかも、プログラムの再利用や移植が容

易なオブジェクト指向情報処理システムを提供することにある。

【0011】

【課題を解決するための手段】上記目的を達成するために、本発明のオブジェクト指向情報処理システムに係る発明の構成は、複数のオブジェクトを含むプロセスが複数存在し、前記オブジェクト間で処理を依頼し合って処理を進めるオブジェクト指向情報処理システムにおいて、各々のオブジェクトに、プロセス内で、オブジェクトを一意に指定するためのオブジェクトアドレスと、このオブジェクト指向情報処理システム全体で、オブジェクトを一意に指定するためのオブジェクト識別子とを設け、各々のプロセス内に、前記オブジェクトと前記オブジェクト識別子の対応を記憶するオブジェクト識別子記憶手段と、外部処理依頼送信手段と、外部処理依頼受信手段とを有し、処理依頼元オブジェクトがその処理依頼元オブジェクトが存在するプロセスとは異なるプロセスに存在する処理依頼先オブジェクトに処理を依頼する場合に、前記外部処理依頼送信手段が、前記オブジェクト識別子記憶手段を参照して、オブジェクトアドレスを、対応するオブジェクト識別子に変換し、そのオブジェクト識別子を指定して、処理依頼メッセージを依頼先プロセスに送信して、前記外部処理依頼受信手段が、外部のプロセスの処理依頼元オブジェクトからの処理依頼メッセージを受信した場合に、前記オブジェクト識別子記憶手段を参照して、前記受信した処理依頼メッセージで指定されたオブジェクト識別子を、対応する処理依頼先オブジェクトのオブジェクトアドレスに変換して、そのオブジェクトアドレスを参照することにより、このメッセージを受信したプロセス内で処理を進めるようにしたものである。

【0012】より詳しくは、上記のオブジェクト指向情報処理システムにおいて、前記外部処理依頼送信手段は、処理依頼メッセージに付加された引数に含まれるオブジェクトアドレスについて、前記オブジェクト識別子記憶手段を参照して、前記オブジェクトアドレスを、対応するオブジェクト識別子に変換して処理依頼メッセージを送信して、前記外部処理依頼受信手段は、前記受信した処理依頼メッセージ中の引数に含まれた前記オブジェクト識別子を、対応する処理依頼先オブジェクトのオブジェクトアドレスに変換して、そのオブジェクトアドレスを参照することにより、このメッセージを受信したプロセス内で処理を進めるようにしたものである。

【0013】さらに詳しくは、前記外部処理依頼送信手段は、処理依頼メッセージに付加された引数に含まれるオブジェクトアドレスについて、前記オブジェクト識別子記憶手段を参照して、前記オブジェクトアドレスを、対応するオブジェクト識別子に変換して処理依頼メッセージを送信して、前記外部処理依頼受信手段は、処理依頼メッセージ中の引数に含まれたオブジェクト識別子の

うち、この処理メッセージを受信したプロセス上に存在するオブジェクトに対応するオブジェクト識別子についてのみ、前記オブジェクト識別子記憶手段を参照して、そのオブジェクト識別子を、対応する処理依頼先オブジェクトのオブジェクトアドレスに変換して、そのオブジェクトアドレスを参照することにより、このメッセージを受信したプロセス内で処理を進めるようにしたものである。

【0014】また、別に詳しくは、上記のオブジェクト指向情報処理システムにおいて、前記オブジェクト識別子は、前記オブジェクトアドレスと同一のデータ型とするようにしたものである。

【0015】

【作用】本発明では、オブジェクトの処理を記述する場合のオブジェクトの指定には、原則として、全てオブジェクトアドレスによるものとして、外部にあるプロセスのオブジェクト場合には、そのオブジェクトアドレスをオブジェクト識別子に変換する。

【0016】したがって、処理依頼元オブジェクトは、処理依頼先オブジェクトが処理依頼元オブジェクトと同一プロセス内に存在するか他のプロセスに存在するかを意識せずに、処理依頼先オブジェクトに処理を依頼し、依頼先オブジェクトは依頼された処理を実行することができるという作用がある。

【0017】したがって、実行時に、プロセスが内にあるか外にあるかの判定が不要になり、コンパイル時で、呼出し先を確定させることが可能になるため、オブジェクトの処理速度に影響を与えずに、オブジェクトの呼出しを高速に処理することできるという作用がある。

【0018】また、アプリケーションプログラマが他のプロセス上の処理依頼先オブジェクトとの通信処理を記述するもなく、処理記述を容易であり、プロセスの如何にかかわらずそのインターフェースを統一的にとり扱え、オブジェクト指向情報処理システムを容易に構築できるといふ作用がある。

【0019】さらに、異なるプロセス上のオブジェクト間の処理依頼はオブジェクト識別子で処理依頼先オブジェクトを指定し、同一プロセス上のオブジェクト間の処理依頼はアドレスで処理依頼先オブジェクトを指定するので、オブジェクト識別子で処理依頼先オブジェクトを指定するプログラムと、アドレスで処理依頼先オブジェクトを指定するプログラムが共存した、オブジェクト指向処理システムを実現でき、プログラムの再利用や移植が容易になるという作用がある。

【0020】

【実施例】以下、本発明の各実施例を、図1ないし図9を用いて説明する。

【0021】〔実施例1〕以下、本発明の一実施例を、図1ないし図8を用いて説明する。

(1) 本発明のシステム構成について

先ず、図1を用いて本発明の一実施例に係るシステム構成について説明しよう。図1は、本発明の一実施例に係るシステム構成をあらわす概念図である。

【0022】図からも明らかなように、本発明の一実施例のシステム構成は、複数の計算機をネットワークで接続した分散システムを想定している。

【0023】すなわち、複数の計算機(図1では、計算機1、計算機2のみを図示している)がネットワーク9に接続されている。そして、計算機1、計算機2には、それぞれ、プロセッサ1p、2p、メモリ1m、2m、ネットワークアダプタ1n、2n、ディスク装置1d、2dが存在している。

【0024】ネットワークアダプタ1n、2nは、ネットワーク9を介して他の計算機と情報をやりとりするための装置である。各計算機にはそれぞれ異なる計算機番号と呼ばれる識別子が与えられており、この計算機番号を指定することにより、他の計算機とネットワークを介した通信が可能である。

【0025】メモリには、計算機が実行するオペレーティングシステム(Operating System、通常「OS」と表記される)のプログラム1o、2o、プロセス11、12、21、22のプログラムが記憶され、これをプロセッサ1p、2pが読みだして処理を実行する。なお、これらプロセスは、「タスク」とも呼ばれることがある。

【0026】OSは、プログラムの実行やユーザとの基本的なサービスを行うソフトウェアである。本発明の一実施例のOSは、複数のプロセスを適当に切り換えながら並行に実行することのできるマルチプロセス機能を有している。例えば、アジソン・ウェスリー社(Addison-Wesley Publish Company)刊、「4.3 BSD UNIX オペレーティング・システムの設計とインプリメント」(1989年)(The Design and Implementation of the 4.3 BSD UNIX Operation System (1989))に記載されているように、現在広く用いられているユニックス(UNIX)・オペレーティングシステム(UNIXはユニックス・システムズ・ラボラトリ(UNIX Systems Laboratory)が開発し、販売しているオペレーティングシステムで、米国その他での登録商標)は、このマルチプロセス機能を有しており、これを利用することにより、本実施例で以下に説明するように、一つの計算機上で複数のプロセスを同時に実行できることになる。

【0027】このユニックス・オペレーティングシステム(以下、単に「UNIX」と記す)では、各プロセスのプログラムは、システムコール(System Call)と呼ばれる手続き呼び出しによりOSに対して、プロセス間通信や、ディスク装置中のファイルへのアクセス処理を依頼することができる。

【0028】プロセス間通信とは、UNIXが提供す

る、プロセス間で情報のやり取りをおこなう手段であって、この機能で通信をおこなえば、複数の計算機にプロセスが分散する形態であっても、送信元のプロセスと送信先のプロセスが、同一計算機上に存在するか、それぞれ別の計算機上に存在するかにかかわらず、同じようにプロセス間の通信をすることが可能となる。また特に、UNIXは、このプロセス間通信機能として、ソケット(socket)と呼ばれる機能を提供している。

【0029】また、プロセスのプログラムには、他のプロセスとの通信を行う通信処理部110、120、210、220、オブジェクトの動作を記述したクラス11c1、11c2、12c1、12c2、21c1、21c2、22c1、22c2等、オブジェクト11o1、11o2、12o1、12o2、21o1、21o2、22o1、22o2等が存在している。

【0030】(2)オブジェクト指向システムにおける基本モデルとオブジェクト間のインタフェースについて次に、図2を用いてオブジェクト指向システムにおける基本モデルとオブジェクト間のインタフェースについて説明しよう。図2は、オブジェクト指向システムにおけるソフトウェア構造の基本モデルをあらわした概念図である。

【0031】オブジェクト指向における「オブジェクト」とは、データとそれに関する手続きから成るプログラムモジュールである。このオブジェクトの種類を「クラス」と呼ぶ。データの値は、各オブジェクト毎に異なるので、オブジェクト単位で記憶する必要がある。一方、手続きは、同一種類のオブジェクトで共通に使用できるのでオブジェクト単位で記憶する必要はない。そこで、手続きは、クラス毎に、データは、オブジェクト毎に記憶する構造を持っている。そして、この手続きは、例えば、C言語では、「関数」と呼ぶプログラムモジュールで記述する。

【0032】したがって、図2に示されるように、クラス11c1は、例えば、関数11c11、11c12、11c13から成る。また、図2で示した例では、オブジェクト11o1、11o2はともに同一クラス11c1に属している。

【0033】オブジェクトのデータは、例えば、C言語では、「構造体」と呼ばれる変数の集合である。図2の例では、オブジェクト11o1には、変数11o11、11o12、11o13が、オブジェクト11o2には変数11o21、11o22、11o23が存在する。同一クラスに属するオブジェクトのデータ構造は同一で、オブジェクトが異なれば、上述のようにその値のみが異なる。

【0034】一般に、オブジェクト指向システムでは、あるオブジェクトが他のオブジェクトに処理を依頼して、オブジェクト同士が通信して処理を進めていく。このようにオブジェクトに処理を依頼するには、オブジェ

クトの手続きに依頼先のオブジェクトと、実行すべき処理とを指定する必要がある。また、必要により、処理に使用するデータを引数として渡す必要がある。

【0035】以下では、オブジェクト間のインタフェースについて詳細に説明していこう。本実施例では、処理依頼時に起動する手続き(関数)はコンパイル時に決定することにする。これは、「静的バインディング方式」と呼ばれる。この処理依頼は、具体的には、実行すべき関数を指定し、処理依頼先のオブジェクトのアドレス(ポインタ)と引数を渡す。

【0036】例えば、C言語の場合、関数名(オブジェクトの構造体へのポインタ、引数1、引数2、・・・)という形式で記述する。

【0037】また、アディソン・ウェスリー社(Addison Wesley)刊、「注釈C++ リファレンス マニュアル」(1990年)(The Annotated C++ Reference Manual (1990))に記載されているC++と呼ばれるオブジェクト指向言語の場合は、処理依頼先オブジェクト、関数名(引数1、引数2、・・・)という形式で記述する。

【0038】C++の方が、C言語よりも表現が洗練されていることが理解されよう。なお、C++ではオブジェクトのクラスの関数は、「メンバ関数」と呼ばれている。

【0039】言語上で関数名で指定した関数呼び出しは、コンパイラを通すことにより、オブジェクトの属するクラスで定義された関数のアドレスを指定したサブルーチンコールになる。呼び出された関数は、オブジェクトのデータである構造体中の変数(メンバ)にアクセスしながら処理を実行する。

【0040】このように、同一プロセスに存在するオブジェクトへの処理依頼は、オブジェクトのアドレス(構造体へのポインタ)を指定した、関数呼び出しにより実行する。しかし、アドレス空間はプロセス毎に異なり、異なるプロセス上のオブジェクトをアドレスにより指定することはできない。

【0041】したがって、分散システム全体でオブジェクトを一意に指定できる識別子が必要である。本実施例では、分散システム内でオブジェクトを一意に指定する識別子として、「オブジェクトID」を用いる。ただし、オブジェクトIDは、それが存在するプロセス以外のプロセスからの処理依頼を受ける可能性のあるオブジェクトのみに与えればよい。すなわち、処理依頼をする対象のオブジェクトがそのプロセスと常に同一のプロセスに属するならば、処理依頼をする対象のオブジェクトにオブジェクトIDを与える必要はないことになる。

【0042】他のプロセスから処理依頼を受ける可能性のあるオブジェクトについては、そのクラス定義中で他

のプロセスに公開することを宣言することにより、オブジェクト中に、オブジェクトIDを記憶する変数を確保する。これは、例えば、C++言語の場合には、他のプロセスに公開するためのクラス（これを、「公開クラス」と呼ぶことにする）を定義しておき、その中で、オブジェクトIDを記憶する変数を宣言しておく。そして、他のプロセスから処理依頼を受ける可能性のあるオブジェクトのクラスを定義する場合には、そのクラス定義中で、公開クラスを継承する（inherit）ことを宣言することにより実現できる。

【0043】オブジェクトにオブジェクトIDを与えるには、プログラム中でオブジェクトIDを指定しておくか、あるいはオブジェクト生成時にオブジェクトIDを与える。生成時にオブジェクトIDを与えるには、例えば、C++言語の場合、公開クラスの「構築子」（Constructor）で、オブジェクトIDを生成して与える処理を記述しておくことにより実現できる。

【0044】（3）本実施例に係る通信処理部の構成、通信処理に用いるデータ構造およびその動作について上述した異なるプロセス上のオブジェクトへの処理依頼は、通信処理部が実行する。よって、次に、図3ないし図7を用いて本実施例に係る通信処理部の構成、通信処理に用いるデータ構造およびその動作について説明することにしよう。図3は、本実施例に係る通信処理部の構成を示したブロック図である。

【0045】まず、この通信処理部110とその各部の動作概要について説明しよう。

【0046】通信処理部110は、遠隔手続き呼び出し処理部111、外部処理依頼送信部112、オブジェクト識別子記憶部113、外部処理依頼受信部114から成る。オブジェクト識別子記憶部113は、外部オブジェクトテーブル1131と、内部オブジェクトテーブル1132から成る。

【0047】オブジェクトが他のプロセス上のオブジェクトへ処理を依頼するには、そのオブジェクトの動作を記述したクラスの手続き（関数）中で、外部処理依頼送信部112を呼び出す。この外部処理依頼送信部112は、外部オブジェクトテーブル1131を参照して、遠隔手続き呼び出し処理部111に他のプロセス上の手続きの呼び出しを依頼する。

【0048】そして、遠隔手続き呼び出し処理部111は、呼び出しに必要な情報を呼び出し先のプロセスに送信する。また、遠隔手続き呼び出し処理部111は、他のプロセスからの呼び出し情報を受信すると、外部処理依頼受信部114を呼び出す。外部処理依頼受信部は内部オブジェクトテーブル1132を参照して、クラスの手続きを呼び出す。これにより異なるプロセス間でオブジェクトの処理依頼が可能になる。

【0049】次に、各部の詳細について説明しよう。先ず、遠隔手続き呼び出し処理部111について説明す

る。

【0050】遠隔手続き呼び出し処理部111は、遠隔手続き呼び出し（Remote Procedure Call、RPC）機能を持つ分散処理環境ソフトウェア、例えば、プレントニス・ホール社（Prentice Hall）刊、「ネットワークコンピューティング システム リファレンス マニュアル」（1990年）（Network Computing System Reference Manual（1990））に記載されているNCSと呼ぶソフトウェアを利用して実現することができる。

10 【0051】遠隔手続き呼び出しでは、呼び出す側（クライアント側）のプロセスのプログラムが、呼び出される側（サーバ側）のプロセスのプログラムの手続き（関数）を、手続き呼び出し（関数呼び出し）と同じ形で呼び出すことができる。この遠隔手続き呼び出しをおこなうには、クライアント側が呼び出すサーバ側の関数の名前や引数名、引数のデータ型等を定義したインタフェース宣言をおこなう必要がある。つまり、このインタフェース宣言をおこなうことにより、呼び出す側（クライアント側）と呼び出される側（サーバ側）との「インタフェース」が確立すると考えるのである。また、この「インタフェース」には、一つまたは複数の関数を含むことができる。そして、定義されたインタフェース宣言をNIDLコンパイラと呼ぶツールに入力することにより、後述するクライアントスタブおよびサーバスタブという遠隔手続き呼び出しに必要なプログラムを生成することができる。

【0052】本実施例では、他のプロセスから処理依頼を受ける可能性のあるクラスについて、それぞれ一つづつ遠隔手続き呼び出しの「インタフェース」を対応させる。また、一つの「インタフェース」には、そのクラスで定義された全ての関数を含めることにする。

【0053】NCSを用いるときには、分散システム全体で一意に識別するための識別子として、UUID（Universal Unique Identifier）と呼ぶ識別子を用いることができる。このUUIDは、UUIDを生成した計算機のアドレスや時刻の情報を含む16バイトの大きさのデータである。上述の遠隔手続き呼び出しのインタフェースを識別するのに、UUID（インタフェースUUIDと呼ぶ）を使用することができる。

40 【0054】遠隔手続き呼び出し処理部111は、遠隔手続き呼び出し実行時ライブラリ1111、クライアントスタブ1112、サーバスタブ1113から成る。スタブとは、遠隔通信のために通信インタフェースのなかだちをする機能を有するものであり、クライアント側にあるものをクライアントスタブ1112、サーバ側にあるものをサーバスタブ1113という。

【0055】このクライアントスタブ1112は、そのプロセスが呼び出す全ての他のプロセス上の手続きについて一つずつ用意される。同様に、サーバスタブは、他のプロセスから呼び出される全ての手続きについて一つ

ずつ用意される。

【0056】遠隔手続き呼び出し実行時ライブラリは、OSのプロセス間通信処理機能を利用して、他のプロセスと通信をおこなう。例えば、UNIXでは、ソケット機能を用いることができる。ソケットは、ソケットアドレスを指定することにより、ポートと呼ばれる通信の口を指定して、プロセス間通信が可能とするものである。本実施例では、プロセスにそれぞれ一つのポートを与え、そのポートのソケットアドレスを指定して遠隔手続きを実行することにより、他のプロセス上の手続きを呼び出すことができる。

【0057】次に、図4および図5を用いてオブジェクト識別子記憶部113について説明する。前述のように、このオブジェクト識別子記憶部113は、外部オブジェクトテーブル1131と内部オブジェクトテーブル1132から成る。図4は、内部オブジェクトテーブルのテーブル構造図である。図5は、外部オブジェクトテーブルのテーブル構造図である。

【0058】図4に示されるように、内部オブジェクトテーブル1132には、他のプロセスから呼び出される可能性のあるオブジェクトについて、オブジェクトIDとオブジェクトのアドレスが記憶される。このオブジェクトIDは、例えば、NC SのUIDを使用することができる。また、オブジェクトIDと、そのオブジェクトが存在するプロセスのソケットアドレスを、分散処理環境ソフトウェアのネームサービスに登録しておく。例えば、前述のNC Sと呼ぶソフトウェアは、ロケーション・ブローカ（Location Broker）と呼ぶネームサービスが提供しており、これを利用する。このネームサービスは、分散処理環境で通信をおこなうに際し、処理対象の名前管理をおこなうために必須のものである。

【0059】また、図5に示されるように、外部オブジェクトテーブル1131には、処理を依頼しようとする他のプロセス上のオブジェクトのオブジェクトIDとそのオブジェクトが存在するプロセスのソケットアドレスが記憶される。すなわち、他のプロセス上のオブジェクトに処理を依頼するプロセスは、その処理を依頼しようとするオブジェクトのオブジェクトIDと、それが存在するプロセスのソケットアドレスを、前述のネームサービスに問い合わせ得ることができ、ネームサービスに問い合わせ得たこのオブジェクトIDとソケットアドレスは外部オブジェクトテーブル1131に記憶するのである。

【0060】ここで、上記のテーブル構造の説明と関連して、オブジェクトの処理の依頼方法について説明すると、一般に、処理依頼元のオブジェクトと処理依頼先のオブジェクトが同一のプロセスにあるときは、処理依頼は、オブジェクトアドレスを指定しておこなえるが、処理依頼元のオブジェクトと処理依頼先のオブジェクトが異なるプロセスにあるときは、この処理依頼が、オブジ

ェクトアドレスを指定することではおこなえない。本発明は、この区別をなくして、処理依頼の形式を統一的に扱うことを可能とするものである。

【0061】そのために、オブジェクトのプログラム中では、処理を依頼するときに、処理依頼先のオブジェクトが、処理依頼元のオブジェクトと異なるプロセスに存在する場合もアドレスで指定する。そして、そのアドレスは、外部オブジェクトテーブル1131の、オブジェクトIDとソケットアドレスが記憶されているエントリのアドレスとする。したがって、他のプロセスに存在するオブジェクトについても、処理を依頼するときに、そのオブジェクトのアドレス（実体は、外部オブジェクトテーブルの1エントリのアドレス）が与えられれば、外部オブジェクトテーブルを参照することにより、オブジェクトIDとソケットアドレスを一意に決定でき、これを用いてプロセス間で通信処理をおこなうのである。

【0062】次に、外部処理依頼送信部112について説明する。この外部処理依頼送信部112は、あるオブジェクトが他のプロセスに存在するオブジェクトに処理を依頼する場合に呼び出される。外部処理依頼送信部112は、外部呼び出し関数によって起動され、依頼元のプロセスのオブジェクトが呼び出す他のプロセス上のオブジェクトのクラスの関数に対応している。またこの外部呼び出し関数は、それぞれ一つずつ、クライアントスタブの関数に対応している。

【0063】上記の様に外部処理依頼送信部の起動は、外部呼び出し関数の呼び出しによりなされる。この呼び出しの仕方は、例えばC言語では、外部呼び出し関数名（オブジェクトの構造体へのポインタ、引数1、引数2、・・・）

という形式である。この第1引数は、オブジェクトのアドレスを示しているが、前述のように、実際には、外部オブジェクトテーブルの1エントリのアドレスである。

【0064】以下では、この外部呼び出し関数の動作を図6の処理フローを用いて説明する。図6は、外部呼び出し関数の動作を示す処理フローである。以下、図6の順を追って説明しよう。

【0065】外部呼び出し関数は、先ず、第1引数で与えられた、処理依頼先のオブジェクトのアドレス、すなわち、外部オブジェクトテーブルのエントリのアドレスから、処理依頼先のオブジェクトのオブジェクトIDと、そのオブジェクトが存在するプロセスのソケットアドレスを得る（処理1121）。また、呼び出しの第2引数以降の引数中にオブジェクトアドレスが存在すればそれをオブジェクトIDに変換する（処理1122）。すなわち、引数中に他のプロセスのオブジェクトのアドレスが存在すれば、外部オブジェクトテーブルを参照してオブジェクトIDに変換し、引数中にそのプロセス中のオブジェクトのアドレスが存在すれば、オブジェクトに記憶されているオブジェクトIDに変換する。

【0066】次に、得たソケットアドレスを指定して、遠隔手続き呼び出しのためのバインディング処理をおこなう(処理1123)。バインディング処理とは、遠隔手続き呼び出しのための通信路確保のための処理である。このバインディング処理の結果、例えば、前述のNCSと呼ぶソフトウェアでは、遠隔手続き呼び出しに必要な、「RPCハンドル」を得る。このRPCハンドルとは、遠隔手続き呼び出しのための情報を密閉したものと考えて良い。そして、遠隔手続き呼び出し処理部のクライアントスタブの関数を呼び出す(処理1124)。呼び出しの仕方は、例えば、上述のNCSの場合は、RPCハンドルを用いて、これをC言語で記述すると、クライアントスタブの関数名(RPCハンドル、オブジェクトID、引数1、引数2、・・・)という形式である。第2引数は処理依頼先のオブジェクトのオブジェクトIDである。これにより、処理依頼先のオブジェクトが存在するプロセスへの処理依頼の遠隔手続き呼び出しがおこなわれる。

【0067】これまで、処理を依頼する側の構成と動作を述べてきたが、今度は逆に、処理を受け取る側の構成と動作を述べていこう。まず、受信側の外部処理依頼受信部114について説明する。

【0068】処理依頼の遠隔手続き呼び出しを受けたプロセスは、サーバスタブを通して、外部処理依頼受信部を呼び出す。外部処理依頼受信部には、サーバスタブの関数全てについて、それぞれ一つずつ内部呼び出し関数が用意され、この関数が呼び出される。内部呼び出し関数は、他のプロセスから呼び出されるオブジェクトのクラスの関数に対応している。内部呼び出し関数の呼び出し方は、例えばC言語では、内部呼び出し関数名(オブジェクトID、引数1、引数2、・・・)という形式である。第1引数は、処理依頼先のオブジェクトのオブジェクトIDである。

【0069】以下では、この内部呼び出し関数の動作を図7を用いて説明する。図7は、内部呼び出し関数の動作を示す処理フローである。以下、図7の順を追って説明しよう。

【0070】内部呼び出し関数は、まず、内部オブジェクトテーブルを検索し、第1引数で与えられた処理依頼先のオブジェクトIDに対応するオブジェクトのアドレスを得る(処理1142)。

【0071】次に、第2引数以降の引数中にオブジェクトIDが存在すれば、それをオブジェクトのアドレスに変換する(処理1142)。すなわち、まず、内部オブジェクトテーブルを検索し、そのオブジェクトIDが存在すれば、このプロセス中にオブジェクトが存在するの

で、引数中のオブジェクトIDを対応するオブジェクトアドレスに変換する。

【0072】内部オブジェクトテーブルに存在しない場合には、このプロセス中にオブジェクトが存在しないので、外部オブジェクトテーブルを検索する。その中に同一のオブジェクトIDを有するエントリが存在すれば、引数中のオブジェクトIDをそのエントリのアドレスに変換する。外部オブジェクトテーブルにも存在しなければ、これまでにこのプロセスでは、参照されなかったということであり、オブジェクトIDとアドレスの変換の外に、外部オブジェクトテーブルへの登録もおこなう。すなわち、具体的には、内部呼び出し関数内の処理で、ネームサービスに問い合わせ、ソケットアドレスを得て、オブジェクトIDとそのソケットアドレスを外部オブジェクトテーブルに追加するとともに、引数中のオブジェクトIDをその追加したエントリのアドレスに変換する。

【0073】最後に、処理依頼先オブジェクトのクラスの関数を呼び出す(処理1143)。呼び出しの仕方は、例えばC言語では、クラスの関数名(オブジェクトへのポインタ、引数1、引数2、・・・)の形式である。また、C++言語の場合では、処理依頼先オブジェクト、関数名(引数1、引数2、・・・)の形式である。

【0074】以上のようにして、本発明の一実施例では、異なるプロセス上に存在するオブジェクト間で処理を依頼し合うことできる。

【0075】(4)本実施例の具体例について

次に、図8を用いて、上で説明してきた本実施例の処理依頼の具体例を説明していくことにしよう。図8は、本発明に係る第一の実施例の処理依頼の動作を具体的に示した模式図である。なお、この図8は、プロセス、オブジェクト間の関係を図示したものであり、図1の他の構成要素は、省略されている。

【0076】計算機1上のプロセス11上にはオブジェクト1101、1102が、計算機2上のプロセス21には、オブジェクト2101が存在しているものとす

る。【0077】それらのオブジェクトID、プロセス11におけるオブジェクトアドレス、プロセス21におけるオブジェクトアドレスは、以下の表1に示される通りであるとする。

【0078】

【表1】

		オブジェクト ID	プロセス11における オブジェクトアドレス	プロセス21における オブジェクトアドレス
プロセス 11	オブジェクト 11o1	I111	A1111	A2111
	オブジェクト 11o2	I112	A1112	A2112 (外)
プロセス 21	オブジェクト 21o1	I211	A1211 (外)	A2211

【0079】表1で、(外)と記したのは、このオブジェクトアドレスが、対応するオブジェクトそのもののアドレスではなく、そのプロセス上の外部オブジェクトテーブル1131中にある対応するオブジェクトのオブジェクトIDとソケットアドレスが記憶されているエントリのアドレスであることを示している。

【0080】さて、ここで、オブジェクト11o1が、外部関数呼び出し112c1、すなわち、m1(A1211, A1112)を実行したとする。正確には、オブジェクトの動作を記述したクラスの手続き(関数)がオブジェクト中に記憶されたデータを読み出してこの処理を実行する。ここで、m1は外部呼び出し関数名である。第1引数のA1211は、処理依頼先のオブジェクトであるオブジェクト21o1のプロセス11におけるオブジェクトアドレスである。第2引数には、オブジェクト11o2のオブジェクトアドレスが指定されている。

【0081】外部関数呼び出しが実行されると、通信処理部110中の外部処理依頼送信部112が起動される。外部処理依頼送信部は外部関数呼び出し中に含まれる引数中のオブジェクトアドレスをオブジェクトIDに変換する。この場合は、第1引数の処理依頼先のオブジェクト21o1のオブジェクトアドレスのA1211をオブジェクトIDのI211に、第2引数のオブジェクト11o2のオブジェクトアドレスのA1112をオブジェクトIDのI112に変換する。このとき、処理依頼先のオブジェクトのソケットアドレスも得る。そして、遠隔手続き呼び出し処理を実行することにより、処理依頼先オブジェクトが存在するプロセス、この場合は、計算機2上のプロセス21への遠隔手続き呼び出しが実行される。

【0082】プロセス21上の通信処理部210が遠隔手続き呼び出しを受けると、外部処理依頼受信部214が起動される。外部処理依頼受信部は、引数中のオブジェクトIDをプロセス21におけるオブジェクトアドレスに変換する。この場合は、第1引数の処理依頼先のオブジェクト21o1のオブジェクトIDのI211をオブジェクトアドレスのA2211に、第2引数のオブジェクト11o2のオブジェクトIDのI112をオブジェクトアドレスのA2112に変換する。そして、処理依頼先オブジェクト、すなわちオブジェクト21o1に対する関数呼び出し214c1、すなわち、

m1(A2211, A2112)

10 を実行する。

【0083】そして次に、このオブジェクト、すなわちオブジェクト21o1が、外部関数呼び出し212c1、すなわち、

m2(A2112)

20 を実行したとする。ここで、m2は外部呼び出し関数名である。第1引数のA2112は処理依頼先のオブジェクトのプロセス21におけるオブジェクトアドレスであり、この場合はオブジェクト11o2が送信先である。第2引数以降の引数は存在しない。この外部関数呼び出しにより、通信処理部210中の外部処理依頼送信部212が起動され、第1引数の処理依頼先のオブジェクトアドレスのA2112をオブジェクトIDのI112に変換する。そして、遠隔手続き呼び出し処理を実行し、計算機1上のプロセス11への遠隔手続き呼び出しが実行される。

【0084】プロセス11上の通信処理部110が遠隔手続き呼び出しを受けると、外部処理依頼受信部114が起動され、引数中のオブジェクトID、すなわちI211をオブジェクトアドレスのA1112に変換する。そして、オブジェクト11o2に対する関数呼び出し114c1、すなわち、m2(A1112)を実行する。

【0085】このように、各計算機上の各プロセス上では、オブジェクトは、そのプロセス上のオブジェクトアドレスを使用して識別する。そして、異なるプロセス上に存在するオブジェクト間で処理を依頼する場合には、オブジェクトアドレスを、システム全体でオブジェクトを一意に決定できるオブジェクトIDに変換して、通信処理(遠隔手続き呼び出し)を実行する。

【0086】(5)本実施例の効果について

本実施例によれば、プロセス内のオブジェクトと他のプロセスのオブジェクトに対して、同じように処理依頼が可能のため、処理の記述が簡単になる。

【0087】プロセス内のオブジェクトの処理依頼については、実行するオブジェクトのクラスの手続き(関数)はコンパイル時に決定されるため、呼ばれたときに、実行する関数をサーチする必要がある場合に比べると、高速に処理がおこなえるという効果がある。

50 【0088】また、本実施例によれば、プロセス内のオ

ブジェクトとプロセス外のオブジェクトを意識することなく、オブジェクト（クラス）のプログラムでは、オブジェクトがどのプロセスに存在するかにかかわらず、オブジェクトアドレスのみでオブジェクトを識別すればよい。そのため、プログラムを開発しやすく、また再利用も容易になるという効果がある。

【0089】さらに、本実施例によれば、プロセス間通信処理において、呼び出しの引数に含まれるオブジェクトアドレスあるいはオブジェクトIDがどのプロセス上のオブジェクトかを意識することなく、オブジェクトアドレスとオブジェクトIDの変換をすればよい。そのため、通信処理部の実現が容易になり、処理も効率化できるという効果がある。

【0090】〔実施例2〕次に、図9を用いて本発明に係る第二の実施例を具体的に説明する。図9は、本発明に係る第二の実施例の処理依頼の動作を具体的に示した模式図である。

【0091】実施例1では、他のプロセスへの処理依頼を実行するときに、処理を依頼する側（クライアント側）では引数に含まれる全てのオブジェクトアドレスをオブジェクトIDに変換し、処理を依頼される側（サーバ側）では引数に含まれる全てのオブジェクトIDをオブジェクトアドレスに変換していた。

【0092】しかし、あるプロセス上において、オブジェクトアドレスが実際にオブジェクトが存在するアドレスを指しているのは、そのプロセス上に存在するオブジェクトのみである。したがって、他のプロセス上のオブジェクトは、オブジェクトアドレスでなくオブジェクトIDで指定しようとするのが本実施例の考え方である。

図9は、この実施例によるオブジェクト間通信の例であり、図8で示した実施例1の具体例と処理の概略は、類似しており、各オブジェクトのオブジェクトアドレスやオブジェクトIDは同様である。

【0093】ただし、この例の場合、他のプロセス上のオブジェクトはオブジェクトアドレスではなく、オブジェクトIDで指定することが実施例1の場合と異なる。

【0094】さて、オブジェクト1101は、外部関数呼び出し112c2、すなわち、

m1 (I211, A1112)

を実行する。第1引数のI211は、処理依頼先のオブジェクトのオブジェクトIDである。第2引数は、オブジェクト1102のプロセス11におけるオブジェクトアドレスである。この呼び出しがされると、外部処理依頼送信部112は、外部関数呼び出し中に含まれる引数中の、プロセス11上に存在するオブジェクトのオブジェクトアドレスをオブジェクトIDに変換する。この場合は、第2引数のオブジェクト1102のオブジェクトアドレスのA1112をオブジェクトIDのI112に変換して、遠隔手続きを実行する。

【0095】プロセス21上の通信処理部210中の外

部処理依頼受信部214は、引数中のオブジェクトIDのうち、プロセス21に存在するオブジェクトのオブジェクトIDのみをプロセス21におけるオブジェクトアドレスに変換する。この場合は、第1引数の処理依頼先のオブジェクト2101のオブジェクトIDのI211をオブジェクトアドレスのA2211に変換する。第2引数のオブジェクトIDは、他のプロセス（すなわち、プロセス11）に存在するオブジェクト1102を示しているため変換されない。そして外部処理依頼受信部214は、オブジェクト2101に対する関数呼び出し214c2、すなわち、
m1 (A2211, I112)
を実行する。

【0096】次に、このオブジェクト、すなわちオブジェクト2101は、外部関数呼び出し212c2、すなわち、

m2 (I112)

を実行する。第1引数のI112は、処理依頼先のオブジェクトのオブジェクトIDである。そして、通信処理部210中の外部処理依頼送信部212が起動されるが、この場合は引数中にオブジェクトアドレスは存在しないのでオブジェクトアドレスからオブジェクトIDへの変換は行わずに、遠隔手続き呼び出し処理を実行する。

【0097】プロセス11上の通信処理部110の外部処理依頼受信部114は、引数中のオブジェクトID、すなわちI211がプロセス11上に存在するオブジェクトを示しているため、これをオブジェクトアドレスのA1112に変換する。そして、オブジェクト1102に対する関数呼び出し114c2、すなわち、

m2 (A1112)

を実行する。

【0098】このように、本実施例では、オブジェクトを指定する場合、そのオブジェクトが存在するプロセスではオブジェクトアドレスで、そのオブジェクトが存在しないプロセスではオブジェクトIDで指定する。これにより、オブジェクトアドレスとオブジェクトIDの変換処理が少なくなり、処理効率が向上するという効果がある。

【0099】〔実施例3〕実施例1および実施例2においては、処理呼び出しの決定方法が、処理依頼に対して実行するオブジェクトのクラスの手続き（関数）をコンパイル時に決定する、いわゆる「静的バインディング」であった。しかし、プログラムの実行時に手続きを探索し決定する、いわゆる「動的バインディング」としてもよい。

【0100】この場合には、処理依頼時に、処理依頼先のオブジェクトがどのクラスかを判定し、そのクラスおよびそのクラスが継承しているクラスの手続きを探索して、実行すべき手続きを決定する。これにより、コンパ

10

20

30

40

50

イル時にはわからず、実行時に動的にクラスが決定されるオブジェクトについても、処理依頼が可能となり、柔軟なシステムを実現できるという効果がある。

【0101】〔実施例4〕上記実施例1ないし実施例3では、オブジェクトアドレスとオブジェクトIDが別のデータ型であることを暗に想定しているような記述をしているが、同一のデータ型としてもよい。

【0102】例えば、実施例1では、アドレス空間が32ビットの計算機の場合、4バイトデータで表現され、オブジェクトIDとしてUUI Dを用いると16バイトデータで表現される。しかし、例えばオブジェクトIDを、使用しないアドレス空間のアドレス識別子として、割り当て表現してもよい。これにより、オブジェクトアドレスとオブジェクトIDが同一のデータ型で表現でき、オブジェクトの指定にオブジェクトアドレスを使用するかオブジェクトIDを使用するかにかかわらず、同じオブジェクトのプログラムを使用でき、プログラムの移植や再利用が容易になるという効果がある。

【0103】

【発明の効果】本発明によれば、複数のプロセスのオブジェクト間で処理を依頼し合う場合において、プロセス内のオブジェクトについてはオブジェクトアドレス指定により高速に処理することでき、効率のよいオブジェクト指向情報処理システムを提供することができる。

【0104】また本発明によれば、上記場合において、アプリケーションプログラマが、他のプロセス上の処理依頼先オブジェクトとの通信処理を行うためのプログラムを記述する必要がなく、複数のプロセス上で通信するオブジェクト指向情報処理システムを、容易に構築することが可能であり、しかも、プログラムの再利用や移植が容易なオブジェクト指向情報処理システムを提供することができる。

【図面の簡単な説明】

【図4】

図 4

オブジェクトID	オブジェクトアドレス
オブジェクトID	オブジェクトアドレス
オブジェクトID	オブジェクトアドレス
オブジェクトID	オブジェクトアドレス
⋮	⋮

* 【図1】本発明の一実施例に係るシステム構成をあらわす概念図である。

【図2】オブジェクト指向システムにおけるソフトウェア構造の基本モデルをあらわした概念図である。

【図3】本実施例に係る通信処理部の構成を示したブロック図である。

【図4】内部オブジェクトテーブルのテーブル構造図である。

【図5】外部オブジェクトテーブルのテーブル構造図である。

【図6】外部呼び出し関数の動作を示す処理フローである。

【図7】内部呼び出し関数の動作を示す処理フローである。

【図8】本発明に係る第一の実施例の処理依頼の動作を具体的に示した模式図である。

【図9】本発明に係る第二の実施例の処理依頼の動作を具体的に示した模式図である。

【符号の説明】

1, 2…計算機、1 p, 2 p…プロセッサ、1 n, 2 n…ネットワークアダプタ、1 d, 2 d…ディスク装置、1 m, 2 m…メモリ、1 o, 2 o…OS、1 1, 1 2, 2 1, 2 2…プロセス、1 1 0, 1 2 0, 2 1 0, 2 2 0…通信処理部、1 1 c 1, 1 1 c 2, 1 2 c 1, 1 2 c 2, 2 1 c 1, 2 1 c 2, 2 2 c 1, 2 2 c 2…クラス、1 1 o 1, 1 1 o 2, 1 2 o 1, 1 2 o 2, 2 1 o 1, 2 1 o 2, 2 2 o 1, 2 2 o 2…オブジェクト、1 1 1…遠隔手続き呼び出し処理部、1 1 2…外部処理依頼送信部、1 1 3…オブジェクト識別子記憶部、1 1 4…外部処理依頼受信部、1 1 3 1…外部オブジェクトテーブル、1 1 3 2…内部オブジェクトテーブル、1 1 1 1…遠隔手続き呼び出し実行時ライブラリ、1 1 1 2…クライアントスタブ、1 1 1 2…サーバスタブ。

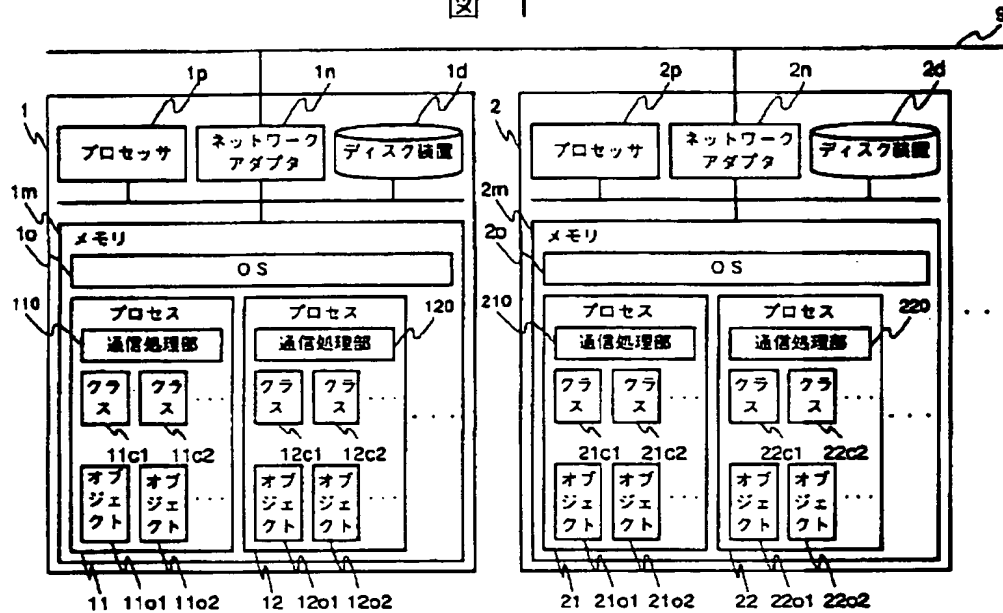
【図5】

図 5

アドレス1	オブジェクトID	ソケットアドレス
アドレス2	オブジェクトID	ソケットアドレス
アドレス3	オブジェクトID	ソケットアドレス
アドレス4	オブジェクトID	ソケットアドレス
⋮	⋮	⋮

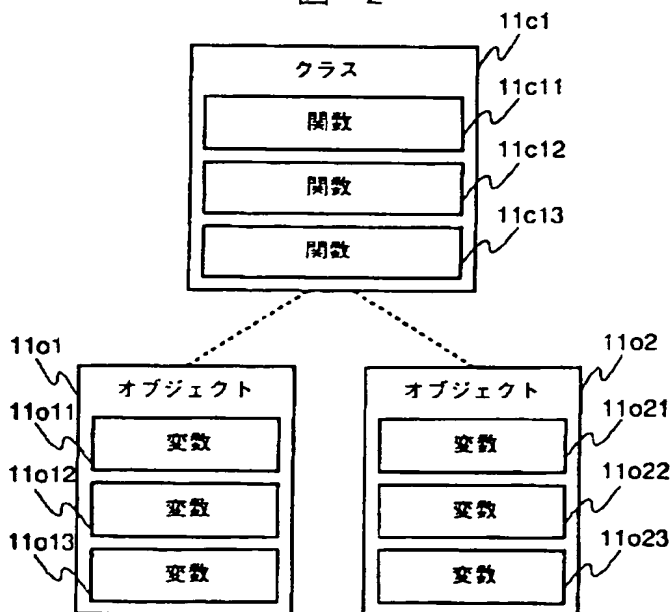
【図1】

図 1



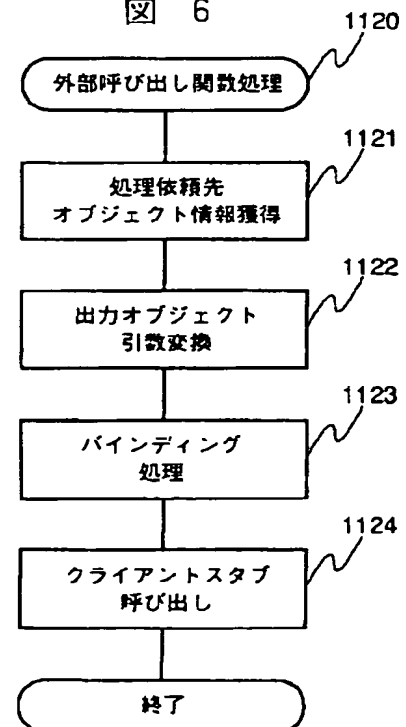
【図2】

図 2



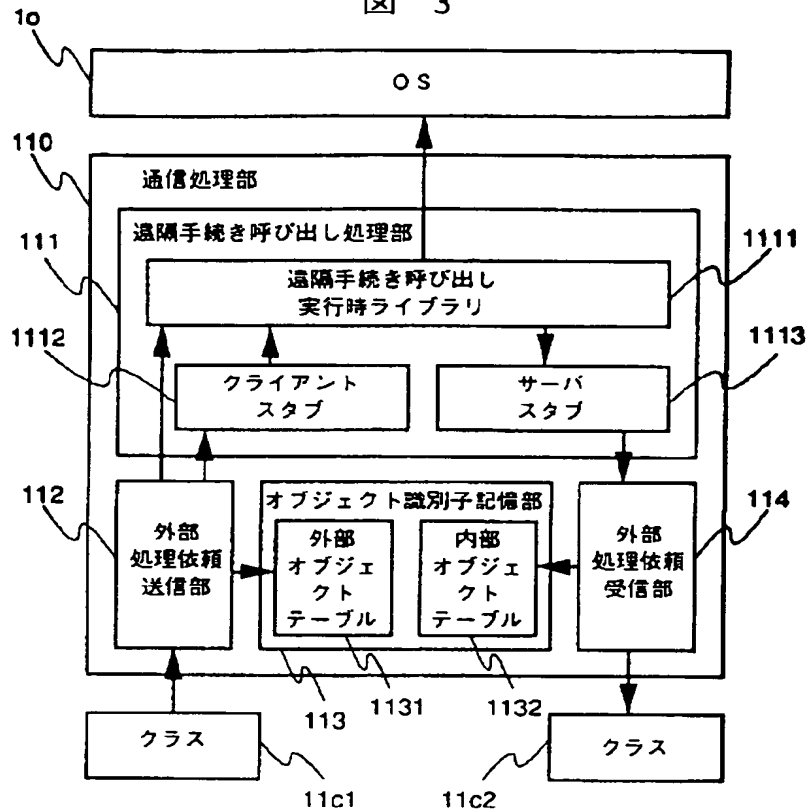
【図6】

図 6



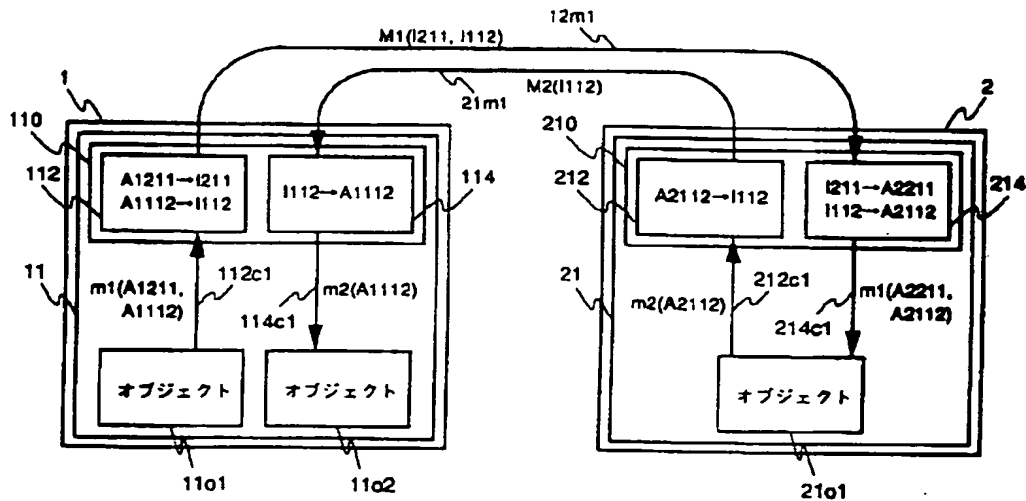
【図3】

図 3



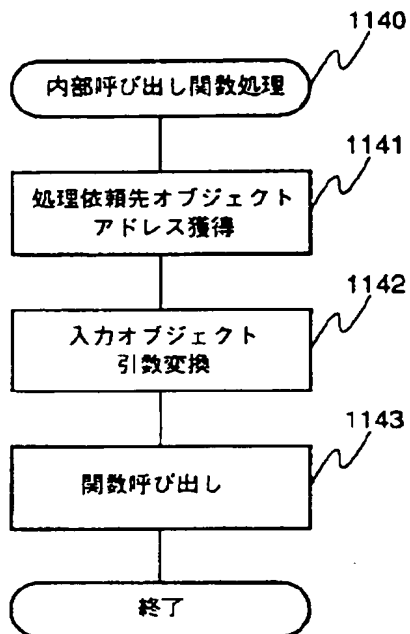
【図8】

図 8



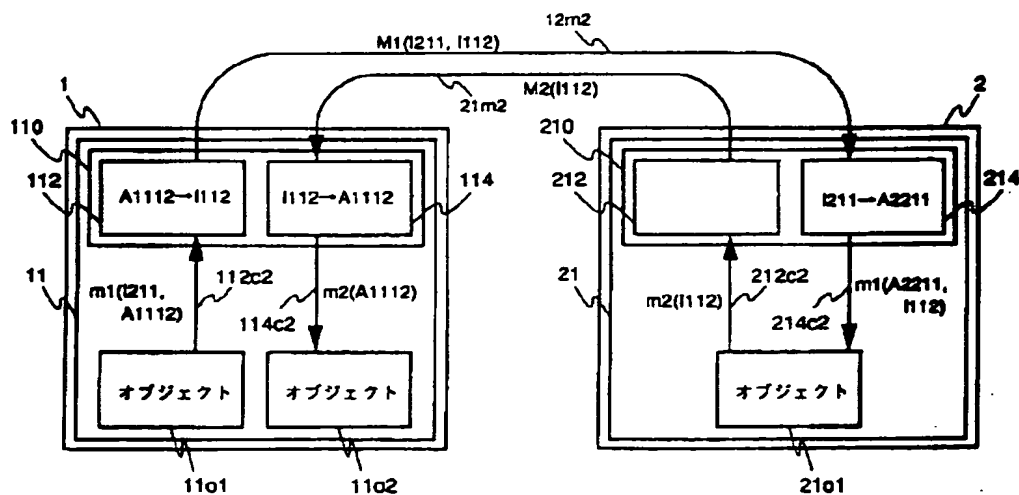
【図 7】

図 7



【図 9】

図 9



フロントページの続き

(72)発明者 恒富 邦彦
茨城県日立市大みか町七丁目 1 番 1 号 株
式会社日立製作所日立研究所内

(72)発明者 中村 智明
茨城県日立市大みか町五丁目 2 番 1 号 株
式会社日立製作所大みか工場内